

# HTML-former

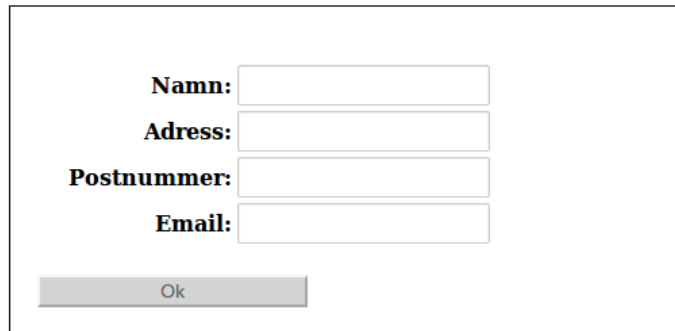
Vi börjar nu att blanda *HTML* med *PHP*. I praktiken, så gör man alltid detta när man bygger hemsidor skrivna i *PHP*. Det som syns - text och bild, knappar och formulär - allt det görs i *HTML*, medan *PHP* ordnar funktionaliteten bakom allt.

## Ett exempel:

Vi ska samla in personuppgifter via ett webbformulär. Hur gör vi?

Först skapar vi utseendet för formuläret. Detta gör vi med *HTML* (vi kommer strax att gå igenom några utav de taggar som används då). Resultatet bli då t.ex. så här:

## Webbformulär




Namn:   
Adress:   
Postnummer:   
Email:   
Ok

Men, eftersom det inte finns något sätt att spara den ifyllda datan på fil med *HTML* (*HTML* är inte ett programmeringsspråk och saknar funktioner för detta), måste vi använda *PHP* för detta.

I webbformuläret finns en OK-knapp,. När man trycker på Ok-knappen, då skickas allt som fyllts i formuläret till en annan sida. Den andra sidan är då ett *PHP*-skript, som vi skapat (exakt hur detta går till, går vi igenom längre ned på sidan).

*PHP*-skriptet tar emot uppgifterna som skickats och ser till att informationen sparas i en databas (vi har förstås lagt in funktioner för databaskommunikationen i *PHP*-skriptet).

## Webbformulär



Namn:   
Adress:   
Postnummer:   
Email:   
Ok

När du trycker på Ok-knappen, körs ett php-skript och informationen som fyllts i HTML-formuläret processeras och sparas i en databas i servern.



Mot slutet av kursen kommer vi att göra just detta. Vi kommer att skapa en enkel blogg, med inloggning och ett sätt att skapa nya inlägg. Inläggen kommer att sparas i en databas.

Men, till en början fokuserar vi enbart på hur vi skapar ett formulär och hur vi kan skicka data från formuläret till ett *PHP*-skript.

## En enkel HTML-form bestående av de nya taggarna `<form>` och `<input>`

Jämför först koden med resultatet.

Så här ser koden ut:

```
<form action="form.php" method="get">
  <b>Namn:</b>
  <input type="text" name="fullname"></input>
  <br>
  <b>Email:</b>
  <input type="email" name="mailaddress"></input>
  <br>
  <input type="submit" value="Ok">
</form>
```

Så här ser resultatet ut:



**Namn:**

**Email:**

Låt oss ta det hela en punkt i taget:

- Allt som syns på bilden ovanför finns inuti taggen `<form>`. Det är alltså vårt HTML-formulär.
- Det finns sammanlagt tre komponenter för HTML-formuläret inom `<form>`. Två textfält och en "Ok-knapp" - alla tre gjorda med taggen `<input>`. Det som skiljer dem åt är attributet `type=""`. Se koden ovanför för att se skillnaden på de tre.
- `<input type="submit">` är en speciell sorts inmatningsmodul. Den är alltså vår "Ok-knapp". När du trycker på "Ok-knappen", då körs det som som är beskrivet med attributet `action=""` innanför taggen `<form>`. I det här fallet har vi `action="form.php"`, vilket betyder att när du trycker på Ok-knappen, då körs filen som heter *form.php*.

- Texten som fyllts i fälten för "Namn" och "Email" skickas vidare till filen *form.php*, med *HTTP*-metoden *GET*. Metoden *GET* är specificerad inom taggen **<form>** med attributet `method="get"`.
- Det sätt vi kan sära på texterna som skickas till filen *form.php*, via *HTTP*-metoden *GET* är med hjälp av attributet `name=""`, som vi har innanför de båda textfälten (de första två **<input>**-taggarna).

## form.php

Nu när vi är klara med *HTML*-formuläret kan vi titta på hur vi gör med *PHP*-delen. Filen som körs, när man trycker på Ok-knappen i *HTML*-formuläret är som sagt *form.php*. Så här ser koden ut för den filen.

```
<?php
$name = $_GET["fullname"];
$email = $_GET["mailaddress"];
echo "Ditt namn är $name och din email är $email.";
?>
```

För att köra *PHP*-skriptet kör man först *HTML*-formuläret (*form.html*), fyller i namn och email och trycker på OK. *HTML*-filen kommer att vidarebefordra till *PHP*-filen och resultatet kommer att se ut t.ex. så här, när *PHP*-filen körts:

Ditt namn är Kaj Ailomaa och din email är kaj.ailomaa@gmail.com.

Variablerna känner vi till sedan innan, liksom hur man skriver ut strängar. Det som är nytt är:

```
$_GET["fullname"];
```

`$_GET` är en inbyggd variabel i *PHP*. Där sparas allt som *PHP*-filen tagit emot via *HTTP*-metoden *GET*. I det här fallet har *PHP*-filen tagit emot två strängar. En under "nyckeln" *fullname* och en under nyckeln *mailaddress*. "Nycklarna" är specificerade i *HTML*-formuläret med attributet `name=""` (kolla gärna *HTML*-koden igen, längre upp, för att se).

## Övning

Skriv av koden för *form.html* och *form.php* och testa sedan att köra *form.html*.